Constraint Dynamics Algorithm for Simulation of Semiflexible Macromolecules

XIONG-WU WU, SHEN-SHU SUNG

The Lerner Research Institute, The Cleveland Clinic Foundation, 9500 Euclid Avenue, Cleveland, Ohio 44195

Received 5 September 1997; accepted 26 May 1998

ABSTRACT: Semiflexible models are often used to study macromolecules containing stable structural elements. Based on rigid body dynamics, we developed a rigid fragment constraint dynamics algorithm for the simulation of semiflexible macromolecules. Stable structural elements are treated as rigid fragments. Rigid fragment constraints, defined as combinations of distance constraints and position constraints, are introduced to limit internal molecular motion to the required mode. The constraint forces are solved separately for each rigid fragment constraint and iteratively until all constraint conditions are satisfied within a given tolerance at each time step, as is done for the bond length constraint in the SHAKE algorithm. The orientation of a rigid fragment is represented by the quaternion parameters, and both translation and rotation are solved by the leap-frog formulation. We tested the algorithm with molecular dynamics simulations of a series of peptides and a small protein. The computation cost for the constraints is roughly proportional to the size of the molecule. In the microcanonical ensemble simulation of polyvalines, the total energy was conserved satisfactorily with time steps as large as 20 fs. A helix folding simulation of a synthetic peptide was carried out to show the efficiency of the algorithm in a conformational search. © 1998 John Wiley & Sons, Inc. J Comput Chem 19: 1555–1566, 1998

Keywords: molecular dynamics simulation; semiflexible models; macromolecules; constraint dynamics; helix folding

Correspondence to: S.-S. Sung; e-mail: sungs@cesmtp.ccf.org

© 1998 John Wiley & Sons, Inc.
CCC 0192-8651 / 98 / 141555-12

Introduction

Molecular dynamics (MD) simulation is a powerful tool for understanding microscopic phenomena, providing kinetic, thermodynamic, and structural information. Its application has been extended to complicated macromolecule systems. The conflict between the complexity of systems and the limit of computation resources demands enhanced simulation efficiency. For example, to accurately describe a protein structure, the positions of thousands of atoms need to be calculated. However, bond lengths and bond angles of protein molecules generally fluctuate rapidly around their equilibrium values. Protein structures, at least their secondary structures, can be well defined by using only ϕ and ψ dihedral angles of their amino acids. For macromolecules, removing some internal degrees of freedom is a major simplification toward increasing efficiency.

Constraining bond length is widely used to reduce degrees of freedom. A constraint dynamics algorithm, the SHAKE algorithm, has been developed by Ryckaert et al.[1] and widely applied. A velocity version, RATTLE, has been proposed by Andersen.[2] These treatments reduce the number of degrees of freedom by about one third and increase the time step by two- to threefold. Following the SHAKE algorithm, several new methods have been developed for constrained dynamics.[3–12] In practice only the bond length constraints have been used widely.[5] When dealing with molecules with many ''triangulated'' units, the convergence could be very slow.[6,7]

Another approach to reducing degrees of freedom is the use of internal coordinates to represent the macromolecular structure. The use of generalized coordinates and Lagrange–Hamilton formalism is the classic approach. However, this approach has only been applied to small molecules of several atoms, such as *n*-butane.[13] Several new methods have been developed using internal coordinates, which have increased the size of the time step up to 30 fs.[14–21] For these internal coordinate approaches, the main difficulties include the analytical complexity and the necessity to invert mass matrices in dynamics equations.

Different models have been used for macromolecular studies. In a rigid model, the internal motion of a molecule is frozen and the whole molecule has only 6 degrees of freedom (5 for linear molecules). A flexible model, in which each atom has 3 degrees of freedom, provides detailed information about local motion within a molecule, but the calculation is time-consuming. For studying conformational changes in a large time scale, removing fast local motions, such as bond length vibrations, can increase the time step significantly. The semiflexible model has fewer degrees of freedom than a flexible model but more degrees of freedom than a rigid model. Many existing MD simulation algorithms for semiflexible models have been developed from flexible models and use constraints between atoms to remove certain degrees of freedom. We call this type of model an atom-based semiflexible model. This approach is suitable for a semiflexible model with a relatively small number of degrees of freedom removed.

When a large portion of degrees of freedom is removed by constraints, the calculation for these constraints will be very time-consuming. In this study, we introduced rigid fragments to replace the many constraints within the stable structural elements. Constraints are used between rigid fragments. We call this type of model a fragment-based semiflexible model. We developed a constraint dynamics algorithm for such a model. This algorithm is described in the next section, followed by a discussion of the properties of the algorithm and examples of its application.

Rigid Fragment Constraint Dynamics

RIGID FRAGMENT SEMIFLEXIBLE MODELS

In this algorithm, a semiflexible model is described as a series of rigid fragments with constraints between them. There are different ways to define the rigid fragments in a macromolecule. Figure 1a shows a model of polyalanine containing rigid fragments. In this model, two types of rigid fragments, one containing atoms C_α and C_β and one containing atom group CONH, are defined and linked with flexible covalent bonds, between C_α and C and between N and C_α, to form a molecule. These rigid fragments are basic moving units, and their motion can be solved using the method of the rigid body dynamic.[22–26] The relative motion between the two neighboring rigid fragments includes the vibrations of the bond length, bond angle, and dihedral angle.

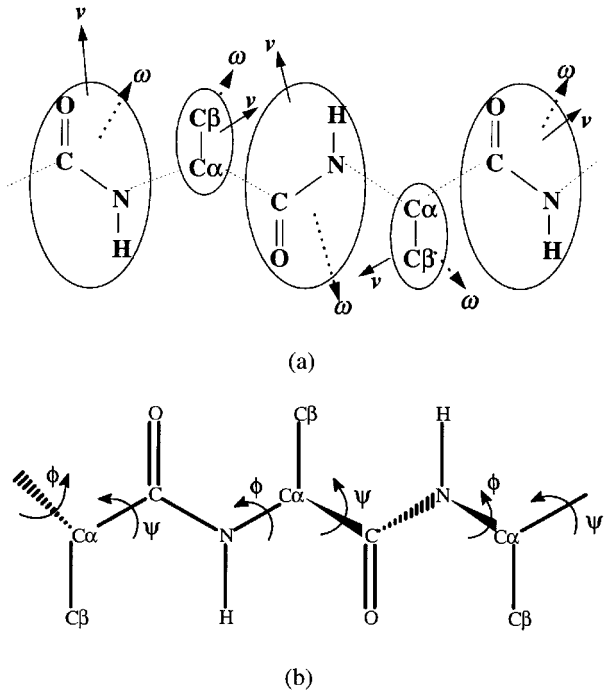


FIGURE 1. (a) A semiflexible polyalanine model consisting of two kinds of rigid fragments, C_α—C_β and CONH. Each rigid body has its own translation, **v**, and rotation, ω. (b) The ϕ, ψ peptide model of polyalanine. In this model all bond lengths, bond angles, and dihedral angles are fixed, except ϕ: C—N—C_α—C and ψ: N—C_α—C—N. The backbone conformation of this model is represented by the ϕ, ψ dihedral angles of the residues.

Figure 1b shows a different semiflexible model of polyalanine. In this model, all bond lengths, bond angles, and dihedral angles in a —CONH— peptide unit are fixed; only the ϕ, ψ dihedral angles are variable (the ϕ, ψ peptide model). We use rigid fragments to represent groups of atoms having relative geometries that are fixed. For example, the ϕ, ψ peptide model for polyalanine shown in Figure 1b has two types of rigid fragments. One contains atom C_α and its covalently bonded neighbor atoms N, C_β, and C. The other contains the peptide group atoms C, O, N, H, and the two attached C_α atoms. In this model, when the ϕ, ψ dihedral angles change, the relative atom positions within the groups remain unchanged. If we use constraints to remove degrees of freedom within the rigid fragment, as in the atom-based semiflexible models, many constraints are ''triangulated,'' such as constraints for the bond angles N—C_α—C, N—C_α—C_β, and C_β—C_α—C, resulting in inefficiency in calculations.

RIGID FRAGMENT CONSTRAINTS

To describe properly the motion of a semiflexible model, the first question is how to achieve the specified flexibility between rigid fragments. For any macromolecule, it is possible to define a large number of different semiflexible models by removing different sets of degrees of freedom. Generally, the degrees of freedom with higher frequencies may be removed when studying slower motions. Therefore, we assume that degrees of freedom are removed in a prescribed order: bond stretching is removed prior to bond bending, and bond bending prior to bond torsion. Under this assumption, in a semiflexible model a possible restricted relative motion between two neighboring rigid fragments can always be described as a combination of two types of constraints: constant distances between two points, each on one fragment; and common atoms shared by the two fragments.

Figure 2 shows several examples of how different constraints result in different flexibility between two rigid fragments. In Figure 2a, two rigid fragments are completely flexible in relation to each other, and no constraint is needed. In Figure 2b, a distance constraint keeps a constant bond length between atom 1N and atom 2C. Here, we denote an atom by the rigid fragment number followed by the atom name. 1N, for example, represents the N atom in rigid fragment 1. In Figure 2c, two rigid fragments share atom C_α, and the bond angles and dihedral angles around C_α are variable. In Figure 2d, two rigid fragments share two atoms, and the dihedral angle around the bond N—C_α is variable.

The two types of constraints can be described as holonomic constraints. We call them the distance constraint (DC) and the position constraint (PC). A DC keeps the distance constant between two points, each on one rigid fragment. A PC superimposes two points, each on one rigid fragment, on each other. These two types of constraints can be expressed as

$$\sigma_c^{DC} = (\mathbf{r}_{jc} - \mathbf{r}_{ic})^2 - d_{ijc}^2 = 0 \quad (1)$$

$$\sigma_c^{PC} = \mathbf{r}_{jc} - \mathbf{r}_{ic} = 0 \quad (2)$$

Here, i and j are the two rigid fragments; c is a constraint index; $\mathbf{r}_{ic}$ and $\mathbf{r}_{jc}$ are the vectors of the constrained points at rigid fragments i and j, respectively; and d_{ijc} is the constraint distance. The PC of eq. (2) is actually three DCs of its

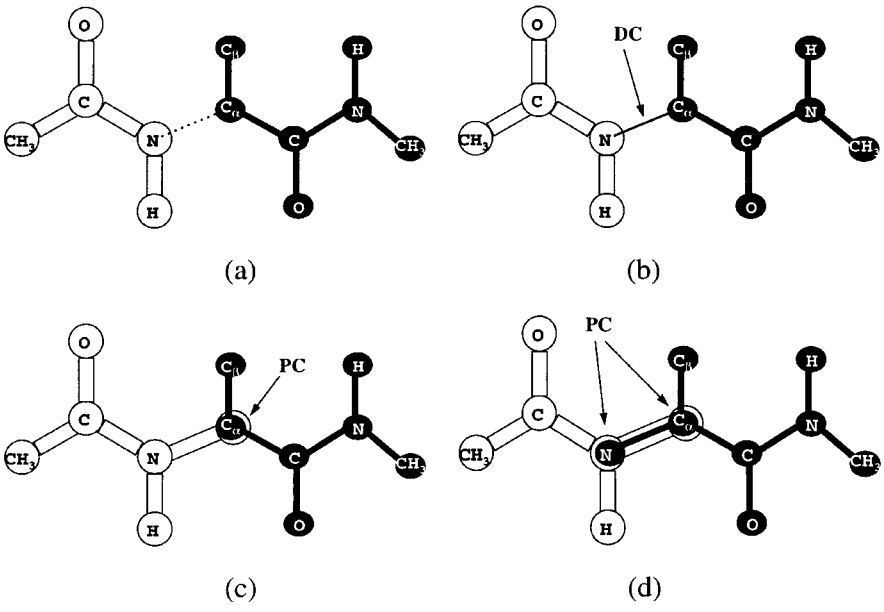


FIGURE 2. Different types of flexibility between a pair of rigid fragments and the rigid fragment constraints for an alanine dipeptide molecule. (a) Two rigid fragments with full flexibility without constraints. (b) Two rigid fragments with a DC representing the fixed bond length N—C_α. (c) Two rigid fragments with a PC at $1C_\alpha/2C_\alpha$. All bond lengths are fixed, and bond angles and dihedral angles around atom $1C_\alpha/2C_\alpha$ are variable. (d) Two rigid fragments with two PCs, one at 1N/2N and the other at $1C_\alpha/2C_\alpha$. All bond lengths and bond angles are fixed, and only the torsion angle around bond N—C_α is variable.

components:

$$\begin{pmatrix} \sigma_c^{\,x} \\ \sigma_c^{\,y} \\ \sigma_c^{\,z} \end{pmatrix} = \begin{pmatrix} x_{jc} - x_{ic} \\ y_{jc} - y_{ic} \\ z_{jc} - z_{ic} \end{pmatrix} = 0 \qquad c \in \mathrm{PC} \qquad (3)$$

A combination of DCs and PCs can limit the relative motion of two rigid fragments to any required mode. We call such a combination of the two types of holonomic constraints the rigid fragment constraint. In Figure 2b, the rigid fragment constraint is a DC between atom 1N and atom $2C_\alpha$. In Figure 2c, it is a PC superimposing atoms $1C_\alpha/2C_\alpha$. In Figure 2d, it has two PCs, one at atoms 1N/2N and one at atoms $1C_\alpha/2C_\alpha$. The ϕ, ψ peptide model of polyalanine shown in Figure 1b consists of two types of rigid fragments, C_α—C(O)—N(H)—C_α and N—$C_\alpha(C_\beta)$—C. The rigid fragment constraints between neighboring rigid fragments are made of two sets of PCs: one at N and C_α for ϕ-angle rotations and the other at C_α and C for ψ-angle rotations. This model is widely used in protein structure studies, and we use it in most examples of this work.

EQUATION OF MOTION

After defining semiflexible models with rigid fragments and rigid fragment constraints, we can derive the constraint forces and describe the motion of the semiflexible models by using rigid body dynamics. In Cartesian space the constraint forces are related to the constraints by eq. (4)[23,24]:

$$\mathbf{f}'_{ic} = \lambda_c \frac{\partial \sigma_c}{\partial \mathbf{r}_i} = -2\lambda_c \mathbf{r}_{ij} \qquad c \in \mathrm{DC} \qquad (4a)$$

$$\mathbf{f}'_{ic} = \lambda_c^x \frac{\partial \sigma_c^{\,x}}{\partial \mathbf{r}_i} + \lambda_c^y \frac{\partial \sigma_c^{\,y}}{\partial \mathbf{r}_i} + \lambda_c^z \frac{\partial \sigma_c^{\,z}}{\partial \mathbf{r}_i} = -\begin{pmatrix} \lambda_c^x \\ \lambda_c^y \\ \lambda_c^z \end{pmatrix} \qquad c \in \mathrm{PC} \qquad (4b)$$

Here, $\mathbf{f}'_{ic}$ is the constraint force on rigid fragment i resulting from holonomic constraint c. λ_c represents the Lagrange multiplier for the constraint. $\mathbf{r}_i$ represents the position of rigid fragment i. In our semiflexible model, each rigid fragment is described by its center of mass $\mathbf{r}$ and its orientation Ω. The equation of motion of rigid fragments can

be separated into two parts: the translation of the center of mass, and the rotation about the center of mass. The constraint forces will affect both the translations and rotations of rigid fragments. The equations of motion for each rigid fragment can be written as:

$$m_i\ddot{\mathbf{r}}_i = \mathbf{f}_i - \sum_j \left\{ \sum_{c\in \mathrm{PC}} \begin{pmatrix} \lambda_{ijc}^x \\ \lambda_{ijc}^y \\ \lambda_{ijc}^z \end{pmatrix} + 2 \sum_{c\in \mathrm{DC}} \lambda_{ijc}\mathbf{r}_{ijc} \right\} = \mathbf{F}_i(\{\lambda\}) \quad (5a)$$

$$\dot{\mathbf{L}}_i = \boldsymbol{\tau}_i - \sum_j \left\{ \sum_{c\in \mathrm{PC}} \mathbf{r}_{ic}^0 \times \begin{pmatrix} \lambda_{ijc}^x \\ \lambda_{ijc}^y \\ \lambda_{ijc}^z \end{pmatrix} + 2 \sum_{c\in \mathrm{DC}} \lambda_{ijc}\mathbf{r}_{ic}^0 \times \mathbf{r}_{ijc} \right\} = \boldsymbol{\Pi}_i(\{\lambda\}) \quad (5b)$$

where $\mathbf{r}_i$, $\mathbf{L}_i$, and m_i are, respectively, the positions of the center of mass, angular momentum, and mass of the rigid fragment i; $\mathbf{f}_i$ and $\boldsymbol{\tau}_i$ are the total nonconstraint force and torque; j represents the neighboring rigid fragments that have rigid fragment constraints with rigid fragment i. c is the index of the holonomic constraint, either a DC or a PC, between rigid fragments i and j. $\mathbf{r}_{ic}^0$ is the vector from the center of mass of rigid fragment i to the constraint point of constraint c on rigid fragment i, and $\mathbf{r}_{ijc}$ is the vector between the two constraint points on the two rigid fragments. For DC c between i and j, the Lagrange multiplier is λ_{ijc}. For PC c, there are three Lagrange multipliers, λ_{ijc}^x, λ_{ijc}^y, and λ_{ijc}^z, for its three components. The generalized force, $\mathbf{F}_i(\{\lambda\})$, and torque, $\boldsymbol{\Pi}_i(\{\lambda\})$, are the combined effect of the nonconstraint forces and constraints.

To solve the equations of motion, eq. (5), we need to solve the Lagrange multipliers. Many methods for this purpose have been proposed.[23] Because a semiflexible system may contain a large number of constraints, a direct calculation of all multipliers will be complicated and time-consuming. As in the SHAKE algorithm,[1] in this work, the Lagrange multipliers were solved iteratively to fulfill the constraint conditions, eqs. (1) and (2), at each time step. In the iterative procedure, each rigid fragment constraint between a pair of rigid fragments was solved successively until all constraint conditions were satisfied to a given tolerance. This approach is suitable for straightforward application to large semiflexible molecules with a complicated set of constraints.

To evaluate the convergence, the following function is used:

$$\Theta_{ij}^2 = \sum_{c\in \mathrm{DC}} \frac{\sigma_{ijc}^2}{4d_{ijc}^2} + \sum_{c\in \mathrm{PC}} \sigma_{ijc}^2$$

Here, the summations are over all holonomic constraints between the two rigid fragments. This function measures how well the constraints between a pair of rigid fragments are satisfied. Only when all the DCs and PCs are fulfilled, does the function Θ take a minimum value of zero.

To illustrate the iterative process in solving the Lagrange multipliers, consider a pair of rigid fragments with a rigid fragment constraint. This rigid fragment constraint is a combination of DCs and PCs. With an iterative procedure, the position of a constraint c on rigid fragment i can be expressed as a perturbation of the position at the preceding iteration step:

$$\mathbf{r}_{ic}(\{\lambda'\}) = \mathbf{r}_{ic}(\{\lambda\}) + \sum_k \frac{\partial \mathbf{r}_{ic}}{\partial \lambda_k}\,\delta\lambda_k$$

Here, we assume that at the preceding iteration step the Lagrange multipliers are $\{\lambda\}$, and, at the present step, they are $\{\lambda'\} = \{\lambda\} + \{\delta\lambda\}$, which fulfills the rigid fragment constraint between i and j. Therefore, for DCs between rigid fragments i and j, the constraint condition, eq. (1), can be rewritten as:

$$\left(\mathbf{r}_{jc}(\{\lambda\}) + \sum_k \frac{\partial \mathbf{r}_{jc}}{\partial \lambda_k}\,\delta\lambda_k - \mathbf{r}_{ic}(\{\lambda\}) - \sum_k \frac{\partial \mathbf{r}_{ic}}{\partial \lambda_k}\,\delta\lambda_k \right)^2 = d_{ijc}^2$$

which leads to the following linear equation by neglecting the terms of the second power of $\delta\lambda$:

$$2\mathbf{r}_{ijc}(\{\lambda\}) \sum_k \frac{\partial \mathbf{r}_{ijc}}{\partial \lambda_k}\,\delta\lambda_k = d_{ijc}^2 - \mathbf{r}_{ijc}^2(\{\lambda\}) \quad (6a)$$

And PC conditions can be rewritten as:

$$\mathbf{r}_{ic}(\{\lambda\}) + \sum_k \frac{\partial \mathbf{r}_{ic}}{\partial \lambda_k}\,\delta\lambda_k = \mathbf{r}_{jc}(\{\lambda\}) + \sum_k \frac{\partial \mathbf{r}_{jc}}{\partial \lambda_k}\,\delta\lambda_k$$

which results in the following linear equation:

$$\sum_k \frac{\partial \mathbf{r}_{ijc}}{\partial \lambda_k}\,\delta\lambda_k = -\mathbf{r}_{ijc}(\{\lambda\}) \tag{6b}$$

Here, $\mathbf{r}_{ijc}(\{\lambda\})$ is the vector from the constraint point on rigid fragment i to the constraint point on rigid fragment j for constraint c. The Lagrange multiplier changes, $\{\delta\lambda\}$, will be solved from these linear equations to arrive at the new values, $\{\lambda'\}$.

It should be noted that, in some cases, not all Lagrange multipliers defined are independent of each other. The number of independent Lagrange multipliers should be the decrease in the number of degrees of freedom in the system. For example, if two rigid fragments are constrained by two PCs, the degrees of freedom decrease by 5. But the number of Lagrange multiplier is $3 \times 2 = 6$. Therefore, eq. (6) provides the same number of equations as or more equations than the total number of independent Lagrange multipliers. There may be more than one set of solutions for these equations. In fact, these different sets of solutions are equivalent, and any set of solutions can be used to integrate the equations of motion. Because these linear equations have fewer than six unknowns, they can be solved very efficiently.

ALGORITHM FOR RIGID FRAGMENT CONSTRAINT DYNAMICS SIMULATION

In this algorithm, rigid fragments are basic moving units. We use rigid body dynamics[23–26] to solve the equations of motion. Rigid fragment positions are described by their centers of mass, and their orientations are described by Evans quaternions,[27] $\mathbf{Q} = (q_0, q_1, q_2, q_3)$, to avoid singularities when solving the equation of motion. To solve the equations of motion, we use the leap-frog formulation for both translations and rotations.[25, 26, 28, 29] For translation, the velocities and positions are solved by the following equations:

$$\dot{\mathbf{r}}_i(t + \tfrac{1}{2}\,\delta t) = \dot{\mathbf{r}}_i(t - \tfrac{1}{2}\,\delta t) + \delta t\,\ddot{\mathbf{r}}_i(t) \tag{7}$$

$$\mathbf{r}_i(t + \delta t) = \mathbf{r}_i(t) + \delta t\,\dot{\mathbf{r}}_i(t + \tfrac{1}{2}\,\delta t) \tag{8}$$

And, for rotation, the following equations are used:

$$\mathbf{L}_i(t) = \mathbf{L}_i(t - \tfrac{1}{2}\,\delta t) + \tfrac{1}{2}\,\delta t\,\dot{\mathbf{L}}_i(t) \tag{9}$$

$$\mathbf{Q}_i(t + \tfrac{1}{2}\,\delta t) = \mathbf{Q}_i(t) + \tfrac{1}{2}\,\delta t\,\dot{\mathbf{Q}}_i(t) \tag{10}$$

$$\mathbf{L}_i(t + \tfrac{1}{2}\,\delta t) = \mathbf{L}_i(t - \tfrac{1}{2}\,\delta t) + \delta t\,\dot{\mathbf{L}}_i(t) \tag{11}$$

$$\mathbf{Q}_i(t + \delta t) = \mathbf{Q}_i(t) + \delta t\,\dot{\mathbf{Q}}_i(t + \tfrac{1}{2}\,\delta t) \tag{12}$$

Beyond conventional rigid body dynamics are rigid fragment constraints. To solve the equations of motion with rigid fragment constraints, one needs to solve the constraints first. As stated earlier, eq. (6a) and (6b) are used to solve the Lagrange multipliers iteratively. In this process, the derivatives of constraint point positions with respect to Lagrange multipliers are the key quantities to be calculated.

The position of each constraint point can be expressed as the position of the center of mass of the rigid fragment and a term related to its orientation:

$$\mathbf{r}_{ic}(\{\lambda\}) = \mathbf{r}_i(\{\lambda\}) + \mathbf{A}_i^{\mathrm{T}}(\{\lambda\})\mathbf{r}_{ic}^{\mathrm{b}}$$

Here, $\mathbf{r}_{ic}$ is the vector of the constraint point on rigid fragment i, and the superscript "b" indicates a vector in the body-fixed frame, which centers at the center of mass of the rigid fragment. $\mathbf{A}$ is the rotation matrix of the rigid fragment, which is related to the quaternions by the following equation[26, 27]:

$$\mathbf{A} = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix} \tag{13}$$

Thus, the derivative of the constraint point position can be separated into a translation part and a rotation part:

$$\frac{\partial \mathbf{r}_{ic}(\{\lambda\})}{\partial \lambda_{\mathrm{k}}} = \frac{\partial \mathbf{r}_i(\{\lambda\})}{\partial \lambda_{\mathrm{k}}} + \frac{\partial \mathbf{A}_i^{\mathrm{T}}(\{\lambda\})}{\partial \lambda_{\mathrm{k}}}\mathbf{r}_{ic}^{\mathrm{b}}$$

These derivatives are related to the generalized forces, $\mathbf{F}_i(\{\lambda\})$, and torques, $\mathbf{\Pi}_i(\{\lambda\})$. For translation, we have:

$$\frac{\partial \mathbf{r}_i(\{\lambda\})}{\partial \lambda_{\mathrm{k}}} = \frac{\partial \mathbf{F}_i(\{\lambda\})}{\partial \lambda_{\mathrm{k}}}\frac{(\delta t)^2}{m_i}$$

The derivative of the rotation matrix can be calculated from the relations among $\mathbf{A}$, $\mathbf{Q}$, ω, and $\mathbf{L}$.

$$\frac{\partial \mathbf{A}_i^{\mathrm{T}}(\{\lambda\})}{\partial \partial \lambda_{\mathrm{k}}} = \frac{\partial \mathbf{A}_i^{\mathrm{T}}(\{\lambda\})}{\partial \mathbf{Q}_i(\{\lambda\})}\frac{\partial \mathbf{Q}_i(\{\lambda\})}{\partial \lambda_{\mathrm{k}}} = \frac{\partial \mathbf{A}_i^{\mathrm{T}}(\{\lambda\})}{\partial \mathbf{Q}_i(\{\lambda\})}\frac{\partial \dot{\mathbf{Q}}_i(\{\lambda\})}{\partial \lambda_{\mathrm{k}}}\,\delta t$$

The derivative $\partial \mathbf{A}_i^{\mathrm{T}}(\{\lambda\})/\partial \mathbf{Q}_i(\{\lambda\})$ can be calculated according to eq. (13). The other derivative is

$$\frac{\partial \dot{\mathbf{Q}}_i(\{\lambda\})}{\partial \lambda_{\mathrm{k}}} = \frac{\partial \dot{\mathbf{Q}}_i(\{\lambda\})}{\partial \omega_i^b(\{\lambda\})} \frac{\partial \omega_i^b(\{\lambda\})}{\partial \lambda_{\mathrm{k}}}$$

The derivative $\partial \dot{\mathbf{Q}}_i(\{\lambda\})/\partial \omega_i^b(\{\lambda\})$ in the above equation can be obtained from the following formula:

$$\dot{\mathbf{Q}} = \begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} 0 \\ \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{pmatrix}$$

And $\partial \omega_i^b(\{\lambda\})/\partial \lambda_{\mathrm{k}}$ can be derived from the relation:

$$\mathbf{L} = \mathbf{I}\,\omega$$

and

$$\mathbf{I}_i \frac{\partial \omega}{\partial \lambda_k} = \frac{\partial \mathbf{L}}{\partial \lambda_k} = \frac{\partial \mathbf{\Pi}(\{\lambda\})}{\partial \lambda_k}\,\delta t$$

From these equations we can calculate the derivatives, $\partial \mathbf{r}_{ijc}/\partial \lambda_k$, at a specific set of $\{\lambda\}$, and from $\partial \mathbf{r}_{ijc}/\partial \lambda_k$ we can solve a new set of multipliers, $\{\lambda'\}$. After we solve $\{\lambda\}$ iteratively, we can solve the equation of motion of our semiflexible systems.

The rigid fragment constraint dynamics algorithm can be summarized as comprising the following steps:

1. Calculate the nonconstraint force acting on each rigid fragment and the motion of the fragment without constraints to obtain the constraint point positions, $\mathbf{r}_{ic}(t + \delta t, \{0\})$.
2. For each rigid fragment constraint, check whether Θ is within the given tolerance. If it is, check the next rigid fragment constraint. If not, go to steps 3 and 4. If all rigid fragment constraints are within the given tolerance, go to step 1 for the next time step.
3. Solve $\{\lambda_k'\}$ from eq. (6).
4. Apply the new constraint forces to the equations of motion and calculate the new position of each constraint point, $\mathbf{r}_{ic}(t + \delta t, \{\lambda\})$. Then go to step 2.

PROGRAMMING AND PARAMETERS

This algorithm was programmed as a package for macromolecule simulation. As in many molecular mechanics programs,[30,31] a molecular structure is represented by the coordinates of its atoms. The atom–atom interaction includes a bonded part and a nonbonded part. The bonded interaction contains bond stretching, bond bending, bond torsion, and improper dihedral torsion. The nonbonded interaction consists of the Lennard–Jones 6-12 potential, the Coulomb interaction, and the hydrogen bonding 10-12 interaction. The united atom parameters of the AMBER force field[32] were used to calculate the interactions. The vacuum dielectric constant $\varepsilon = 1$ was used for the Coulomb interaction. All nonbonded atom pairs were considered in our calculation, and cutoff was not used. The 1-4 nonbonded interactions were scaled down by a factor of 0.5, as suggested for the AMBER parameter.[32]

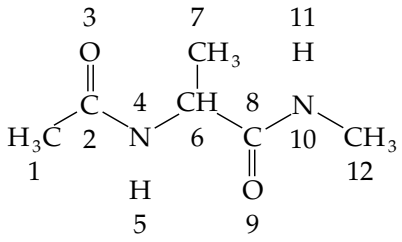
A semiflexible model of a macromolecule is defined by its rigid fragments and the DCs and PCs between these fragments. The forces and torques acting on each rigid fragment are calculated from the forces acting on its atoms. If an atom is shared by several rigid fragments, its mass and interaction forces are divided among them. In this work, we evenly divided the mass of a shared atom and the force acting on the atom among the rigid fragments. For constant temperature simulations, we used the constraint methods developed by Brown and Clarke[33] and Fincham et al.[34] to scale the velocities and angular momenta of rigid fragments to maintain the system at a given temperature. A macromolecule can be studied using different semiflexible models by defining different sets of rigid fragments and rigid fragment constraints.

Compared with the conventional MD simulation software for flexible macromolecule models, this program needs several more arrays to store the positions, orientations, velocities, and angular momenta of rigid fragments. At each time step, an iteration process is needed to solve the equations of motion. Because the atoms in the same rigid fragment are fixed relative to one another, the trajectory of a semiflexible system may be recorded by the centers of mass and the orientations of rigid fragments. For semiflexible models with large rigid fragments, such a trajectory file helps to save disk space.

Properties of the Algorithm and Its Applications

The rigid fragment constraint dynamics algorithm has been tested on an alanine dipeptide for

the convergence of the iteration. Various semiflexible models can be defined for this molecule, as shown in Table I. The atoms in this molecule are labeled 1–12:



We applied our algorithm to the nine models in Table I to test the convergence of the iteration. The MD simulations were conducted at $T = 300$ K with the time step of $\Delta t = 10$ fs. The Θ values under various rigid fragment constraints are shown in Table I. The iteration converged rapidly for all rigid fragment constraints. With a tolerance of 0.0005 Å, commonly used when applying the SHAKE algorithm, one or two iterations in our algorithm are enough to reach the convergence. With the same number of degrees of freedom, PCs usually lead to a faster convergence than DCs, as shown in models 3 and 4, or in models 8 and 9. In models 6 and 7, the convergence is about the same.

A comparison between the atom-based constraint and the fragment-based constraint was conducted on simulations of the ϕ, ψ peptide model of an alanine dipeptide. If only the ϕ, ψ dihedral angles are variable, we need to fix all bond lengths, bond angles, and other dihedral angles. The fully flexible molecule has 36 degrees of freedom, whereas a ϕ, ψ peptide model of the dipeptide has only 8 degrees of freedom. Therefore, 28 internal coordinates need to be fixed. Using the atom-based constraints, we fixed all the 11 bond lengths, 10 bond angles, 5 improper dihedral angles, and 2 dihedral angles around the peptide bonds. With 28 atom-based constraints, the simulation was very time-consuming. A 1-ps simulation took 1.2 min on an SGI Iris 4D35 workstation. Using fragment-based constraints, this semiflexible model was represented as three rigid fragments: I, atoms 1–6; II, atoms 4 and 6–8; and III, atoms 6 and 8–12. Between I and II, two PCs at atoms 4 and 6 were applied, and between II and III, two PCs at atoms 6 and 8 were applied. This 1-ps simulation took only 0.015 min with the same constraint tolerance. Furthermore, as shown in what follows, with rigid fragment constraints, a larger time step can be used in a simulation.

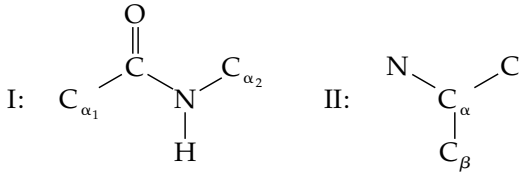
For a series of rigid fragment constraints between each pair of neighboring rigid fragments, one constraint force affects other constraints during the iteration, and more iterations are needed than that for a single constraint. However, constraints placed far apart along a chain molecule have little effect on each other. Thus, once a molecule reaches a certain size, the number of iterations will not increase with the molecule size. To examine the relation between the computation cost and molecular size, we simulated polyvalines Ac—V_n—NH_2 with $n = 16$, 32, 64, and 128. The ϕ, ψ peptide model was used. Two types of rigid fragments were defined for the backbone of the

TABLE I.
Convergence in the Iterations to Solve Rigid Fragment Constraints.

						Θ (Å) at each iteration step				
No.	RF1[a]	RF2[b]	DC[c]	PC[d]	N_f[e]	0	1	2	3	4
1	A1	B1	4–6	—	11	0.20e-2	0.18e-5	0.22e-9	0.27e-13	
2	A1	B1	2–6/4–6	—	10	0.20e-1	0.29e-3	0.57e-6	0.18e-8	0.59e-11
3	A1	B1	2–6/4–6/5–6	—	9	0.34e-1	0.46e-3	0.85e-6	0.34e-8	0.15e-10
4	A2	B1	—	6	9	0.37e-1	0.58e-4	0.75e-7	0.18e-9	0.41e-12
5	A2	B1	4–7	6	8	0.42e-1	0.28e-3	0.73e-6	0.18e-8	0.14e-10
6	A2	B1	4–7/4–8	6	7	0.61e-1	0.51e-3	0.11e-5	0.51e-8	0.12e-10
7	A2	B2	—	4/6	7	0.16e0	0.79e-3	0.35e-5	0.34e-8	0.14e-10
8	A2	B2	5–7	4/6	6	0.16e0	0.14e-2	0.23e-3	0.15e-5	0.17e-6
9	A3	B2	—	4/6/7	6	0.29e0	0.18e-2	0.86e-5	0.12e-7	0.86e-10

[a] RF1 denotes the first rigid fragment. A1 contains atoms 1–5, A2 contains atoms 1–6, and A3 contains atoms 1–7.
[b] RF2 denotes the second rigid fragment. B1 contains atoms 6–12, and B2 contains atoms 4 and 6–12.
[c] The atom pairs between which DCs are applied.
[d] The atoms at which PCs are applied.
[e] N_f is the number of degrees of freedom of the system. It is 12 if no rigid fragment constraint is applied.

peptides:



The two consecutive rigid fragments were constrained by two PCs at the nitrogen (or the carbonyl carbon) and the α-carbon atoms. Similar to the backbone rigid fragment I, the end groups were defined as rigid fragments and constrained by two PCs with their neighboring backbone rigid fragment II.

Each valine side chain was represented by a rigid fragment, C_α—$C_\beta(C_\gamma)_2$, connected to the backbone fragment II by two PCs at C_α and C_β. The reason for choosing a homopolymer is to make the molecular size proportional to the number of residues. Polyvaline is a branched chain molecule and is a more general model than polyalanine. These simulations were carried out at the temperature $T = 300$ K with a compact initial structure. Table II shows the CPU times consumed by the constraint calculation (with a tolerance of 0.0005 Å) and by the nonbonded interaction in a 10-ps simulation on an SGI Indigo-2 R10000 computer. The CPU time for the iteration is approximately a linear function of the molecular size, but the CPU time for nonbonded interaction calculation is approximately proportional to the second power of the molecular size. As the molecular size increases, the CPU time for the nonbonded interaction increases much faster than for the iteration. Reducing the nonbonded interaction calculation is crucial for computational efficiency, and it may be achieved by using a larger time step.

Table II also shows the CPU time required with various time steps. Although the number of iterations increased with the time step, the combined CPU time of the constraint iteration and the nonbonded interaction decreased as time steps increased, before reaching a minimum at about 10 fs (depending on the size of the peptide). The optimized time step depends on both the molecular system and constraint tolerance. A larger system will require more CPU time for the nonbonded interaction calculation, and a larger time step is preferred. A smaller tolerance will result in a higher computation load for the constraints, and a smaller time step is preferred.

To further test the constraint dynamics algorithm, a small protein, bovine pancreatic trypsin inhibitor (BPTI), has been simulated. Because the MD simulation of protein folding is currently not possible, the native initial structure is used in the simulation. The BPTI molecule contains 58 residues with three disulfide bonds between residues 5 and 55, 14 and 38, and 30 and 51, respectively. The semiflexible model of the BPTI allowed single bond rotations only. A rigid fragment on the side chain of each cysteine contains both bonded sulfur atoms and the rotation of the disulfide bond is allowed. The maximum time step depends on the semiflexible model, not on the algorithm itself. BPTI contains many different types of amino acids, and many small fragments have to be defined for the side chains to allow all the single bond rotations. Because these small fragments move fast, time steps of 20 fs or larger cannot be used. Table II shows the CPU times of the BPTI simulations with time steps from 2 to 10 fs. With the time step of 10 fs, the combined CPU time of both the nonbonded and constraint calculations is the lowest.

As mentioned in the Introduction, the calculation of the SHAKE type of constraints converges very slowly for molecules with many ''triangulated'' units.[6,7] In our method, these units may be defined as rigid fragments, such as fragment II of the peptide backbone with the side chain C_β atom. Within a rigid fragment, the ''triangulated'' constraints are not needed. Therefore, the branched

TABLE II.
CPU Time (Seconds) for Nonbonded Interaction (NB) and Constraint (CONS) Calculations (Tol. = 0.0005 Å).

	2 fs		5 fs		10 fs		20 fs	
Time step	NB	CONS	NB	CONS	NB	CONS	NB	CONS
V16	2.03	5.54	0.76	7.21	0.38	9.65	0.19	12.32
V32	7.73	11.65	3.08	15.83	1.54	19.99	0.80	25.00
V64	30.97	24.12	12.39	33.65	6.17	43.69	3.15	53.67
V128	125.67	48.22	62.88	85.92	25.07	89.87	12.52	108.62
BPTI	39.50	15.81	15.78	24.01	7.96	30.03	—	—

chain does not pose new problems in the algorithm.

In the ϕ, ψ peptide model, the high-frequency motion of the covalent bond stretching and bending is eliminated. Because rigid fragments are generally heavier than single atoms, they move more slowly than atoms at any given temperature. Therefore, larger time steps are feasible. A simple check for the validity of large time steps is to examine the energy conservation in a microcanonical ensemble. We did microcanonical ensemble MD simulations for a 16-residue polyvaline. The constraint tolerance was 0.000005 Å. These simulations were carried out at $T = 300$ K with a compact initial structure. Table III shows the energies of the molecule and their root-mean-square deviations during simulations with time steps from 2 to 30 fs, together with the total energy drifts calculated from 10-ps microcanonical ensemble simulations. The fluctuations in total energy and the total energy drifts increased as the time step increased. At a time step of 30 fs, they increased dramatically. The rigid fragment constraint dynamics simulations with a time step of 20 fs or less showed satisfactory conservation in total energy. The size of time steps depends on the molecular model used in a simulation. Generally, a more flexible model contains faster moving small fragments, and needs smaller time steps.

Recent developments related to internal coordinate (or torsion angle) methods have shown the advantage of allowing large time steps, such as 20 or 30 ps.[12, 17, 18, 21] Our method reached similar time steps in the simulation of polyvaline. The computation cost of our method is roughly proportional to the number of the interaction centers, which is similar to the advantage shown in some recent efforts.[21]

The relation between the computation load and tolerance has been tested on a 16-residue peptide. The relation is roughly a linear function. A larger tolerance results in a smaller computation load. This result does not mean that we should choose a large tolerance in a simulation. The choice depends on the percentage of computing time needed for the constraint in a simulation. Because the computing time for the nonbonded interaction increases much faster with molecular size, for large systems, the constraint calculation takes a small portion of the CPU time. In this case, using a small tolerance does not slow down a simulation significantly. On the contrary, for small systems, in which calculating constraints takes most of the CPU time, increasing tolerance can speed up a simulation significantly.

To study the efficiency of the rigid fragment constraint dynamics in a conformational search, we simulated folding of a synthetic peptide: Ac—AAQAAAAQAAAAQAAY—NH_2, where A, Q, and Y represent alanine, glutamine, and tyrosine, respectively. Circular dichroism measurements showed that this peptide has a helical content of about 50% in aqueous solution.[35, 36] The backbone structure of this synthetic peptide is the same as polyalanine, and the ϕ, ψ peptide model was used. The extra degrees of freedom on the side chains of residues Q and Y make the molecule a branched chain molecule. The side chain of a glutamine was described by a rigid fragment, $C_\gamma C_\delta(O_\delta)N_\delta$, which was constrained with backbone rigid fragment II by three distance constraints, between C_α and C_γ, C_β and C_γ, and C_β and C_δ, so that only the dihedral angles X—C_α—C_β—X, X—C_β—C_γ—X and X—C_γ—C_δ—X in the side chain were variable. For the side chain of tyrosine, two rigid fragments were defined: $C_\beta C_6 O_\zeta$ and $C_\zeta O_\zeta H$. $C_\beta C_6 O_\zeta$ was constrained with the backbone rigid fragment II by a PC at C_β and a DC between C_α and C_γ. $C_\zeta O_\zeta H$ was constrained with $C_\beta C_6 O_\zeta$ by two PCs at C_ζ and O_ζ, respectively. Only the dihedral angles X—C_α—C_β

TABLE III.
Energies of 16-Polyvaline in Various 10-ps NVE Simulations (Tol. = 0.000005 Å).

Time steps (fs)	E_{tot} (kcal/mol)		E_{kin} (kcal/mol)		E_{pot} (kcal / mol)		$\Delta E_{tot}/\Delta t$ (kcal/mol/ps)
	Ave.	RMSD	Ave.	RMSD	Ave.	RMSD	
2	−382.076	0.027	18.509	3.663	−400.585	3.660	−0.001
5	−381.954	0.043	20.455	3.371	−402.410	3.374	0.004
10	−381.679	0.083	20.216	3.326	−401.895	3.344	0.024
20	−380.667	0.339	21.001	3.612	−401.668	3.699	0.062
30	−370.387	5.506	26.627	5.481	−397.014	4.697	15.39

—X, X—C_β—C_γ—X, and X—C_ζ—O_ζ—H_ζ in the side chain were variable.

The simulation was conducted at 300 K with a fully extended initial conformation. The time step was 20 fs. The conformational changes during the simulation are shown in Figure 3, in which the helix segments are represented by vertical lines at the positions of corresponding residues. A residue is defined as helical when its ϕ, ψ angles are within 30° of the standard value for the α-helix, $\phi = -57°$ and $\psi = -47°$.[37] As the simulation started, the peptide relaxed from the extended conformation. At about 30 ps, a helix turn formed at residues 9–11 (Fig. 4). It propagated in both directions, forming a helical segment at residues 7–14 at about 40 ps. A separate helical turn formed at the N-terminus at about 60 ps. At about 200 ps, the two helical segments joined, forming a complete helix. The residues 15 and 16 deviated frequently from a helical conformation, showing that the C-terminus of the helix was more flexible than the N-terminus. This simulation showed that the algorithm is efficient in sampling the conformational space and finding the low-energy conformation.

During this simulation, the semiflexible peptide folded into a complete helix in hundreds of picoseconds. The time scale of folding depends on the potential functions of the force field. Without solvent, the interactions are usually exaggerated, making the conformation change faster, but the exaggerated interaction can also trap the molecule in the local energy minima before reaching the lowest energy conformation. Solvent can also have a damping effect on folding. Discussion about the force field and the helix-folding mechanism is beyond the scope of this work. Some helix-folding simulations using the Monte Carlo (MC) or the conventional MD method have been described elsewhere.[38–41]

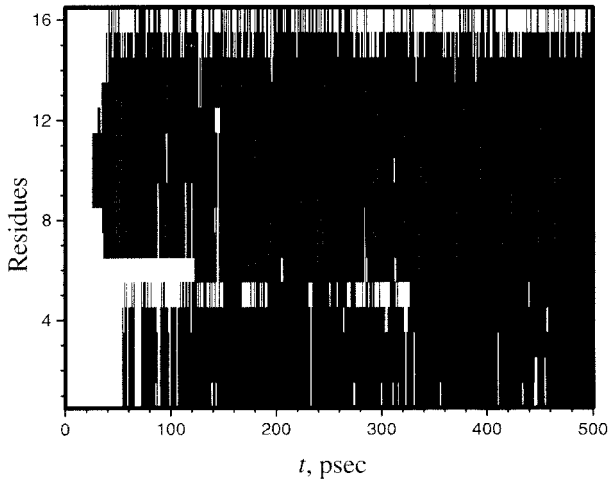


FIGURE 3. The helix segment location of the 16-residue synthetic peptide during the folding simulation. The conformations at each picosecond were analyzed, and the helical residues are shown as vertical lines at their corresponding positions. The simulation started from a fully extended conformation. The simulation temperature was 300 K and the time step was 20 fs. The tolerance for constraints was 0.0005 Å.

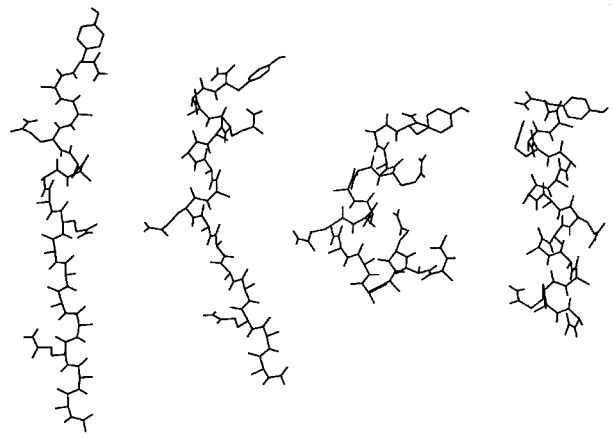


FIGURE 4. Selected conforrnations of the 16-residue synthetic peptide in the folding simulation. From left to right the conformations are at 30, 40, 60, and 500 ps.

Summary

A rigid fragment constraint dynamics algorithm has been developed for the simulation of semiflexible macromolecules. Stable structural elements are treated as rigid fragments, and rigid fragment constraints are introduced to achieve specific relative motions between rigid fragments. The rigid fragment constraint consists of two kinds of holonomic constraints: distance constraints and position constraints. This algorithm is based on rigid body dynamics and works in the Cartesian coordinate system. The rigid fragment constraints are solved iteratively, as is done for bond length constraints in the SHAKE algorithm. Test simulations with this algorithm showed a satisfactory convergence in solving rigid fragment constraints. For large molecules with many rigid fragment constraints, the computation cost for the constraints is roughly a linear function of the molecular size. With the ϕ, ψ peptide model, energy conserved satisfactorily during the simulation of various polyvalines

with time steps as large as 20 fs. The application of this algorithm has been tested on synthetic peptide folding. The efficiency of simulating the conformational change of peptides makes this algorithm a promising tool for studying protein folding in the future.

References

1. J.-P. Ryckaert, G. Ciccotti, and H J. C. Berendsen, *J. Comput. Phys.*, **23**, 327 (1977).
2. H. C. Andersen, *J. Comput. Phys.*, **52**, 24 (1983).
3. G. Ciccotti, M. Ferrario, and J.-P. Ryckaert, *Mol. Phys.*, **47**, 1253 (1982).
4. J.-P. Rychaert, *Mol. Phys.*, **55**, 549 (1985).
5. W. F. van Gunsteren and M. Karplus, *Macromolecules*, **15**, 1528 (1982).
6. W. F. van Gunsteren and H. J. C. Berendsen, *Mol. Phys.*, **34**, 1311 (1977).
7. W. F. van Gunsteren, *Mol. Phys.*, **40**, 1015 (1980).
8. D. J. Tobias and C. L. Brooks, *J. Chem. Phys.*, **89**, 5115 (1988).
9. J. Durup, *J. Phys. Chem.*, **95**, 1817 (1991).
10. D.-S. Bae and E. J. Haug, *Mech. Struct. Mach.*, **15**, 359 (1987).
11. D.-S. Bae and E. J. Haug, *Mech. Struct. Mach.*, **15**, 481 (1987).
12. L. M. Rice and A. T. Brunger, *Prot. Struct. Funct. Genet.*, **19**, 277 (1994).
13. K. Refson and G. S. Pawley, *Mol. Phys.*, **61**, 669 (1987).
14. A. K. Mazur and R. A. Abagyan, *J. Biomol. Struct. Dyn.*, **6**, 815 (1989).
15. R. A. Abagyan and A. K. Mazur, *J. Biomol. Struct. Dyn.*, **6**, 833 (1989).
16. A. K. Mazur, V. E. Dorofeev, and R. A. Abagyan, *J. Comput. Phys.*, **92**, 261 (1991).
17. K. D. Gibson and H. A. Scheraga, *J. Comput. Chem.*, **11**, 468 (1990).
18. K. D. Gibson and H. A. Scheraga, *J. Comput. Chem.*, **11**, 487 (1990).
19. A. Jain, N. Vaidehi, and G. Rodriguez, *J. Comput. Phys.*, **106**, 258 (1993).
20. A. Jain, *J. Guidance Control Dyn.*, **14**, 531 (1991).
21. A. M. Mathiowetz, A. Jain, N. Karasawa, and W. A. Goddard III, *Prot. Struct. Funct. Genet.*, **20**, 227 (1994).
22. K. Nicklas, P. Bopp, and J. Brickmann, *J. Chem. Phys.*, **101**, 3157 (1994).
23. J. Garcia de Jalon and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems*, Springer, Berlin, 1994.
24. E. A. Desloge, *Classical Mechanics, Vol. 2*, Wiley, New York, 1982.
25. D. Fincham, *CCP5 Quarterly*, **2**, 6 (1981).
26. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, 1987.
27. D. Evans, *Mol. Phys.*, **34**, 317 (1977).
28. R. W. Hockney, *Comput. Phys.*, **9**, 136 (1970).
29. D. Potter, *Computational Physics*, Wiley, New York, 1972.
30. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.*, **4**, 187 (1983).
31. D. A. Pearman, D. A. Case, J. C. Caldwell, G. L. Seibel, U. C. Singh, P. Weiner, and P. A. Kollman, *AMBER 4.0*, University of California, San Francisco, CA, 1991.
32. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, and P. Weiner, *J. Am. Chem. Soc.*, **106**, 765 (1984).
33. D. Brown and J. H. R. Clarke, *Mol. Phys.*, **51**, 1243 (1984).
34. D. Fincham, N. Quirke, and D. J. Tildesley, *J. Chem. Phys.*, **84**, 4535 (1986).
35. J. M. Scholtz, E. J. York, J. M. Stewart, and R. L. Baldwin, *J. Am. Chem. Soc.*, **113**, 5102 (1991).
36. J. M. Scholtz, H. Qian, V. H. Robins, and R. L. Baldwin, *Biochemistry*, **32**, 9668 (1993).
37. V. Daggett and M. Levitt, *J. Mol. Biol.*, **223**, 1121 (1992).
38. S.-S. Sung, *Biophys. J.*, **66**, 1796 (1994).
39. S.-S. Sung, *Biophys. J.*, **68**, 826 (1995).
40. S.-S. Sung and X.-W. Wu, *Prot. Struct. Funct. Genet.*, **25**, 202 (1996).
41. S.-S. Sung and X.-W. Wu, *Biopolymers*, **42**, 633 (1997).